

PORTFOLIO

애자일 기반 건강관리 서비스 개발 프로젝트

이름	박희수
생년월일	1996.06.09
전화번호	010-4145-6593
주소	경기도 의왕시 행복마을3로 18
이메일	liaparkel@gmail.com
GitHub	https://github.com/liaparkel/portfolio

PORTFOLIO

CONTENTS

01. 프로필

02. 팀 프로젝트 개요 1) 업무 소개
 2) 아키텍처 변화

03. Spring Boot 기반 1차 스프린트

04. REST API 기반 2차 스프린트

05. MSA 기반 3차 스프린트

06. 최종 산출물

07. 회고

01. 프로필

2025.02 ~2025.07	한국소프트웨어 기술진흥협회	Java 기반 DevOps 개발자 양성 훈련과정 수료
2022.05 ~2023.11	한미정밀화학	SOP(표준작업지침서)에 따른 제품 품질시험 수행 URS(사용자요구사항규격서) 문서 관리, 시험 성적서 작성 식약처, FDA, PMDA, INFARMED 등 국제기관 실사 준비 및 자료 대응
2021.01 ~2021.06	한양대학교에리카 산학협력단	DNA PCR 증폭용 프라이머 염기서열 빅데이터 분석 데이터/메타데이터 표준화를 통한 데이터 디지털화 및 업로드 '바이오데이터엔지니어양성사업' 교육과정 이수
2015.03 ~2020.02	단국대학교	생명과학과 학사 졸업

02. 업무 소개



Selfit은 스스로 **계획하고 기록하며**,
건강한 습관을 만들어 가는 사람들을 위한 **동기부여 커뮤니티 플랫폼**입니다.

2025.06.13~2025.06.20

REST API 리팩터링

1차 스프린트

2차 스프린트

3차 스프린트

2025.05.12~2025.06.01

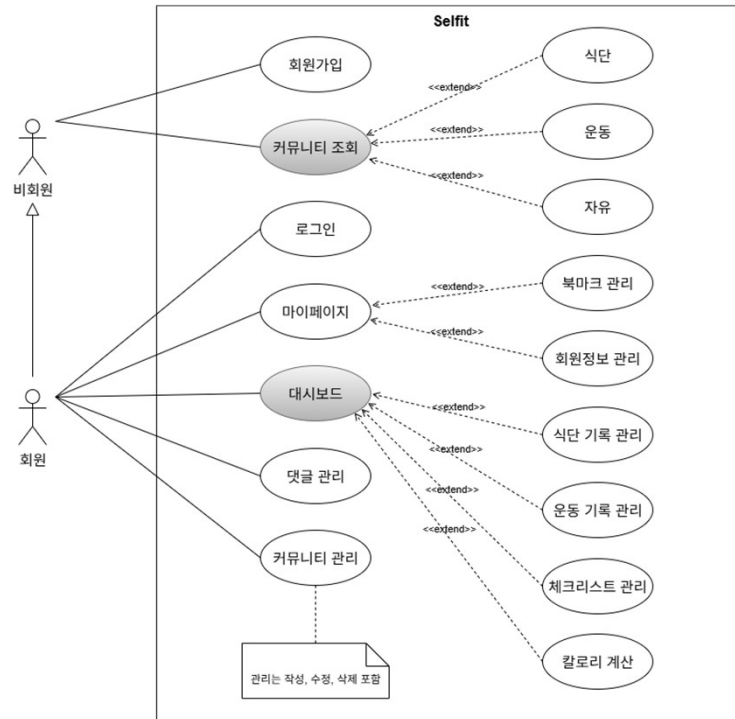
대시보드 식단 기록 기능 구현

2025.06.30~2025.07.15

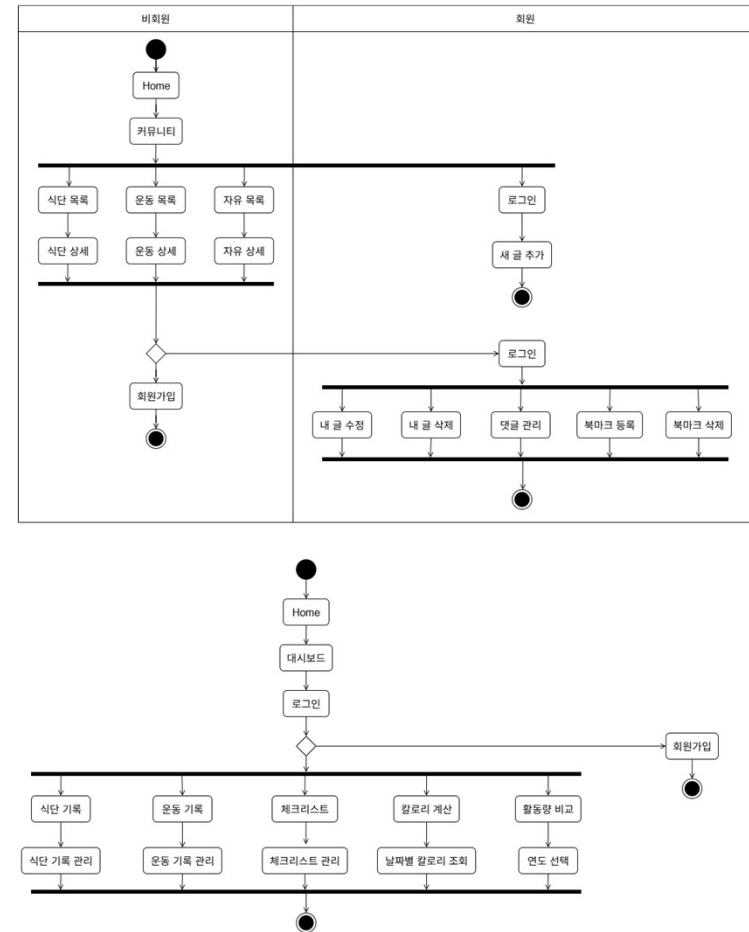
MSA 리팩터링

02. 업무 소개

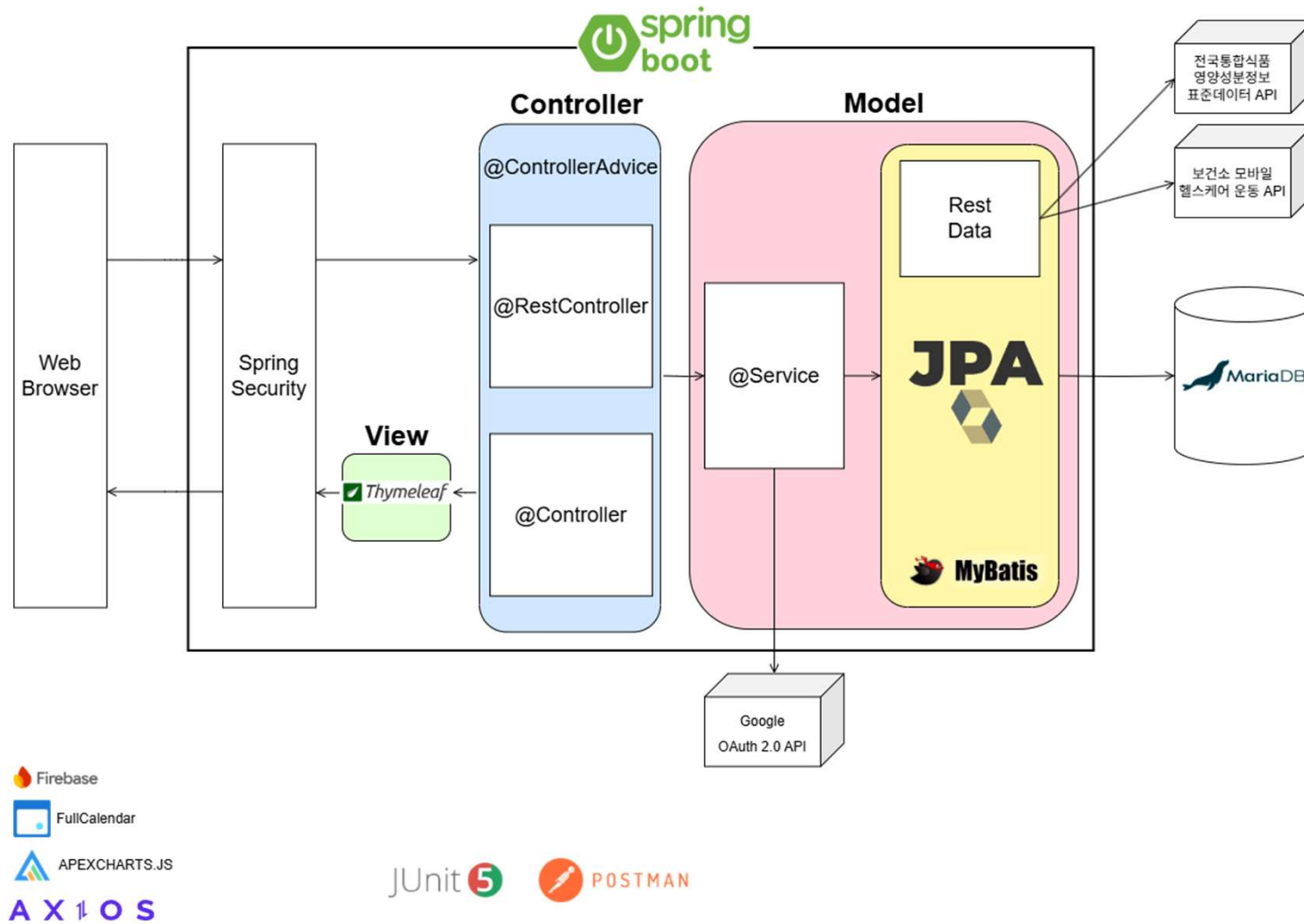
UseCase Diagram



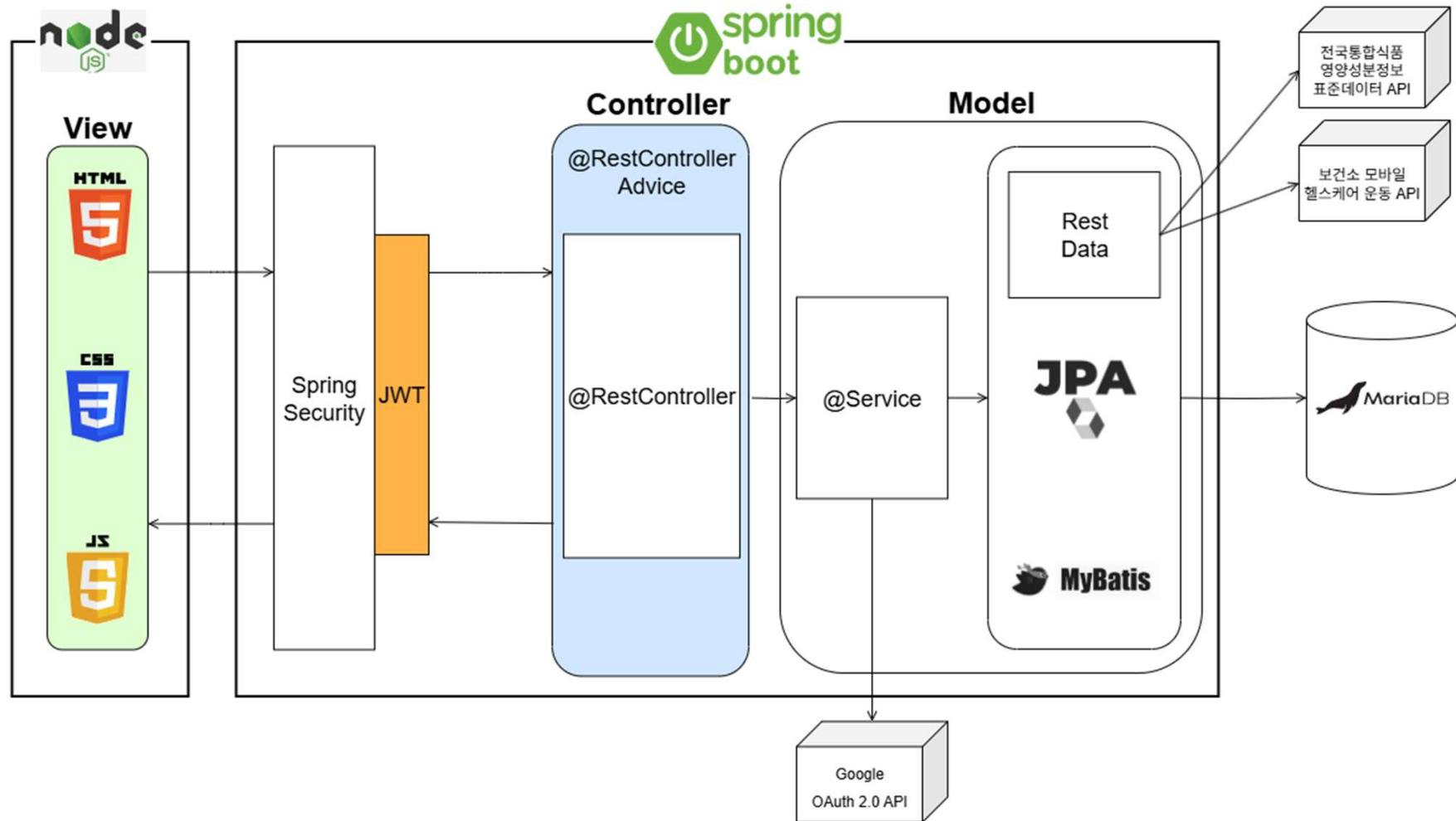
Activity Diagram



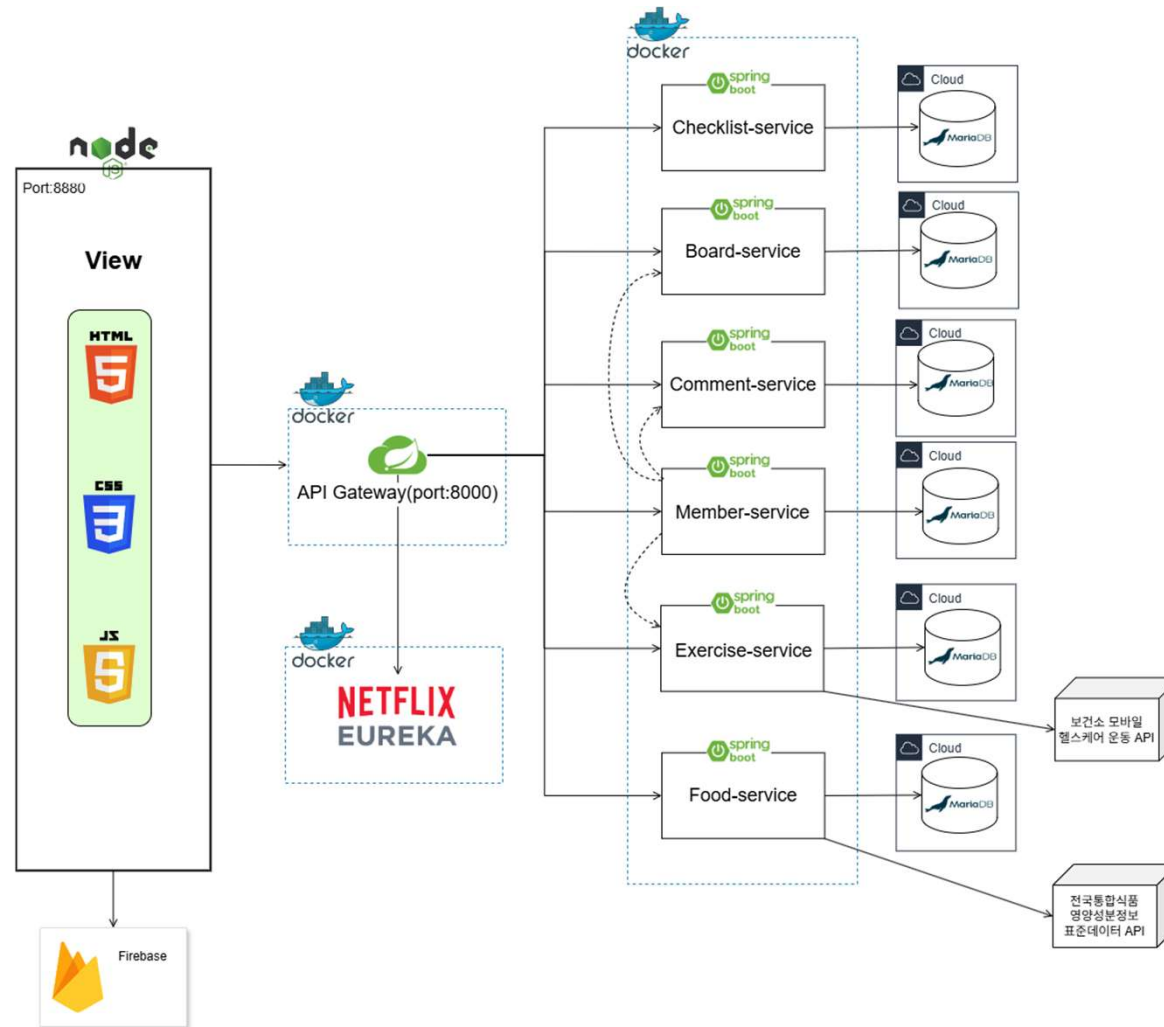
02. 아키텍처 변화 – 1차 스프린트



02. 아키텍처 변화 – 2차 스프린트



02. 아키텍처 변화 – 3차 스프린트



03. Spring Boot 기반 1차 스프린트

기존 구조의 한계

객체 생성과 의존성 관리가 직접 이루어져 중복과 결합도가 높고, 테스트가 어려움 → IoC 컨테이너와 DI가 가능한 **Spring Boot** 도입

시행착오

사용자 기록 입력 시 음수, 중복 등의 예외 상황에 대한 검증 누락

MyBatis에 익숙해 JPA 도입에 대한 기술적 거부감 존재

해결방법

사전 유효성 검증 및 예외 응답 로직 추가

일부 기능에 JPA 제한적으로 도입

04. REST API 기반 2차 스프린트

기존 구조의 한계

백엔드와 프론트엔드가 한 서버 내에서 실행되어 코드를 수정하면 전체 서버 재가동 → 서버 분리로 결합도를 낮춰 독립적인 개발과 배포 가능

Controller와 Thymeleaf 사용으로 백엔드와 프론트엔드 의존성 문제 발생 → REST API 적용하여 독립적인 개발 환경 구축

시행착오

프론트와 서버 분리로 기존 세션 기반 로그인 방식이 동작하지 않음

해결방법

JWT 기반 인증 구조를 설계해 사용자 인증 흐름을 토큰 중심으로 구현

05. MSA 기반 3차 스프린트

기존 구조의 한계

기존 모놀리식 구조에서 다른 사람의 코드가 내 코드에도 영향을 주는 문제 발생 → 마이크로 서비스로 나누어 영향을 최소화하고 독립성 확보

특정 서비스 장애 시 전체 시스템을 확장해야 하는 비효율성 → 트래픽이 많은 서비스만 선택적으로 인스턴스를 추가 확장하도록 구조 변경

시행착오

마이크로 서비스로 나누니 서비스 간 직접적인 호출 불가능

해결방법

RestTemplate을 사용하여 JSON 기반의 HTTP 통신 방식으로 데이터 연동

06. 최종 산출물 – Swagger

OpenAPI definition v0 OAS 3.0
[/v3/api-docs](#)

Servers

food-controller ^

PUT	/api/food-service/member/{memberId}	▼
POST	/api/food-service/member/{memberId}	▼
DELETE	/api/food-service/member/{memberId}	▼
POST	/api/food-service/open-search	▼
POST	/api/food-service/kcal/year/member/{memberId}	▼
POST	/api/food-service/kcal/member/{memberId}	▼
POST	/api/food-service/foods/member/{memberId}	▼
POST	/api/food-service/avg/year/member/{memberId}	▼
GET	/api/food-service/api-test	▼

06. 최종 산출물 – Eureka

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
APIGATEWAY-SERVICE	n/a (1)	(1)	UP (1)
BOARD-SERVICE	n/a (1)	(1)	UP (1)
CHECKLIST-SERVICE	n/a (1)	(1)	UP (1)
COMMENT-SERVICE	n/a (1)	(1)	UP (1)
EXERCISE-SERVICE	n/a (1)	(1)	UP (1)
FOOD-SERVICE	n/a (1)	(1)	UP (1)
MEMBER-SERVICE	n/a (1)	(1)	UP (1)

06. 최종 산출물 – 업무 상세 설계서

member-service													
번호	업무	흐름	URL	MemberController			Exception	MemberService			MemberRepository		권한
				종류	req	res	res	method	input	output	input	output	
1	개인정보 조회	마이페이지 들어갈 때 요청	/api/member-service/member/{memberId}	GET	@PathVariable String memberId	ResponseEntity<ResGetMember>	-	getMember	memberId	MemberDto	memberId	MemberEntity	ROLE_ANONYMOUS
2	이메일 중복체크	회원가입 페이지에서 이메일 중복체크 누를 때 요청	/api/member-service/check-email	POST	@RequestBody ReqCheckEmail reqCheckEmail	ResponseEntity<ResMessage>	-	checkEmail	memberDto	Boolean	email	MemberEntity	ROLE_ANONYMOUS
3	닉네임 중복체크	회원가입 페이지에서 닉네임 중복체크 누를 때 요청	/api/member-service/check-nickname	POST	@RequestBody ReqCheckNickname reqCheckNickname	ResponseEntity<ResMessage>	-	checkNickname	memberDto	Boolean	nickname	MemberEntity	ROLE_ANONYMOUS

food-service													
번호	업무	흐름	URL	FoodController			Exception	FoodService			FoodRepository		권한
				종류	req	res	res	method	input	output	input	output	
1	일별 섭취칼로리 조회	칼로리 계산에서 해당 날짜를 클릭하면 총섭취칼로리 조회	/api/food-service/kcal/member/{memberId}	POST	@PathVariable String memberId @RequestBody ReqGetIntakeKcal reqGetIntakeKcal	ResponseEntity<ResGetIntakeKcal>	IllegalArgumentException, NullPointerException	getIntakeKcal	FoodDto	List<FoodDto>	memberId, intakeDate	List<FoodEntity>	ROLE_USER
2	연도별 섭취칼로리 조회	대시보드 식단에서 식단 그래프에 연도별 섭취칼로리 차트 출력	/api/food-service/kcal/year/member/{memberId}	POST	@PathVariable String memberId, @RequestBody ReqGetYearIntakeKcal reqGetYearIntakeKcal	ResponseEntity<List<ResGetYearIntakeKcal>>	IllegalArgumentException, NullPointerException	getYearIntakeKcal	FoodDto	List<FoodDto>	memberId, year	List<FoodEntity>	ROLE_USER
3	섭취 상세 조회	대시보드 식단에서 특정 날짜 클릭 시 해당 날짜에 맞는 기록 조회	/api/food-service/foods/member/{memberId}	POST	@PathVariable String memberId @RequestBody ReqGetFood reqGetFood	ResponseEntity<List<ResGetFood>>	IllegalArgumentException, NullPointerException	getFood	FoodDto	List<FoodDto>	memberId, intakeDate	List<FoodEntity>	ROLE_USER
4	섭취 음식 리스트 추가	대시보드 식단에서 기록 추가	/api/food-service/member/{memberId}	POST	@PathVariable String memberId @RequestBody ReqAddFood reqAddFood	ResponseEntity<ResMessage>	DataIntegrityViolationException	addFood	FoodDto	FoodDto	foodEntity foodTotalEntity memberId, localDate	FoodEntity FoodTotalEntity FoodTotalEntity	ROLE_USER
5	섭취 음식 리스트 삭제	대시보드 식단에서 기록 삭제	/api/food-service/member/{memberId}	DELETE	@PathVariable String memberId @RequestBody ReqRemoveFood reqRemoveFood	ResponseEntity<ResMessage>	IllegalArgumentException	removeFood	FoodDto	-	memberId, FoodId memberId, localDate foodTotalEntity	int FoodTotalEntity FoodTotalEntity	ROLE_USER
6	섭취 음식 항목 수정(섭취량)	대시보드 식단 패널에서 그 항목에 해당하는 섭취량 수정	/api/food-service/member/{memberId}	PUT	@PathVariable String memberId @RequestBody ReqSetFood reqSetFood	ResponseEntity<ResMessage>	IllegalArgumentException	setFood	FoodDto	FoodDto	memberId, FoodId memberId, intakeDate	FoodEntity foodTotalEntity	ROLE_USER
7	연도별 섭취 칼로리 평균 조회	대시보드 요약에서 섭취량 필터 누르기	/api/dashboard-service/avg/year/member/{memberId}	POST	@PathVariable String memberId @RequestBody ReqGetYearIntakeAvgAll reqGetYearIntakeAvgAll	ResponseEntity<List<ResGetYearIntakeAvgAll>>	NullPointerException	getYearIntakeAvgAll	FoodDto	List<FoodDto>	Map<String, Object>	List<Map<String, Object>>	ROLE_USER
8	음식 검색	식단 기록에서 패널을 열고 먹은 음식 검색	/api/food-service/open-search	POST	@RequestBody ReqOpenFoodSearch reqOpenFoodSearch	ResponseEntity<Mono<List<ResOpenFoodSearch>>>	-	getFoodByNameLike	FoodApiDto	Mono<List<FoodApiDto>>	pageNo numOfRows	Mono<List<FoodApiDto>>	ROLE_USER

06. 최종 산출물 – UI 페이지

The screenshot displays a web application interface for a checklist. The browser address bar shows the URL: `54.180.249.146:8880/html/dashboard/checklist.html`. The application header includes the 'Selfit' logo, a user profile icon, and a '로그아웃' (Logout) button.

The main content area is titled '체크리스트' (Checklist) and shows a calendar for '2025년 6월' (June 2025). The calendar grid includes days from 1st to 12th. A sidebar on the left contains navigation links: '대시보드' (Dashboard), '커뮤니티' (Community), and '마이페이지' (My Page).

Overlaid on the right side of the screen is a network inspector tool. The 'Response' tab is active, displaying a JSON array of checklist items. The items are structured as follows:

```
[{"memberId": 0, "checklistId": 1, "checkId": 1, "checkDate": null, "checkContent": "바벨 스위트 100개", "isChecked": 0}, {"memberId": 0, "checklistId": 1, "checkId": 2, "checkDate": null, "checkContent": "물 2L 마시기", "isChecked": 0}, {"memberId": 0, "checklistId": 1, "checkId": 3, "checkDate": null, "checkContent": "오", "isChecked": 0}, {"memberId": 0, "checklistId": 1, "checkId": 4, "checkDate": null, "checkContent": "오", "isChecked": 0}, {"memberId": 0, "checklistId": 1, "checkId": 5, "checkDate": null, "checkContent": "오", "isChecked": 0}]
```

07. 회고

서버 환경 설정 파일 관리의 복잡함

모놀리식 구조에서는 단일 env 파일로 모든 환경 설정을 관리했지만, 서비스가 분리되면서 각 서비스마다 개별 환경 설정 파일로 관리했습니다. 하지만 흩어진 설정 파일들의 관리가 어려웠습니다.

따라서 **Config Server**를 도입하여 하나의 중앙 집중식 서버에서 환경 설정을 관리할 필요성을 느꼈습니다.

마이크로 서비스 분리 기준

기존 모놀리식에서 업무 방식 자체가 바뀌기 때문에 마이크로 서비스를 어떤 기준으로 나눌지 정하기 어려웠습니다.

팀 내 논의를 통해 기존 RestController 기준으로 업무 크기를 확인하여 분리하기로 결정했지만, 이 과정에서 많은 시간이 소요되었습니다.

이를 통해 MSA에서 업무를 나누는 것에 대한 **사전 계획과 논의가 중요**하다는 것을 경험했습니다.

감사합니다
